Development of Robust Adaptive Inverse models using Bacterial Foraging Optimization

H Pal Thethi
Dept. of Electronics & Communication Engg., KIIT University, Bhubaneswar, India
Email: harpalnitr@gmail.com
Babita Majhi and Ganapati Panda
Dept. of IT, Institute of Technical Education and Research, S 'O' A University, Bhubaneswar, India
School of Electrical sciences, Indian Institute of Technology Bhubaneswar, India
Email:babita.majhi@gmail.com/ganapati.panda@gmail.com

***Abstract*— Adaptive inverse models find applications in communication and magnetic channel equalization, recovery of digital data and adaptive linearization of sensor characteristics. In presence of outliers in the training signal, the model accuracy is severely reduced. In this paper three robust inverse models are developed by recursively minimizing robust norms using BFO based learning rule. The performance of these models is assesses through simulation study and is compared with those obtained by standard squared norm based models. It is in general, observed that the Wilcoxon norm based model provides best performance. Moreover the squared error based model is observed to perform the worst.**

***Index Terms*— Adaptive inverse models, Robust norms, Robust adaptive inverse model, Bacterial foraging optimization**

I. Introduction

In digital data communication systems, high speed data transmission over band limited channel often causes inter-symbol interference (ISI) due to adverse effect of dispersive channel [1]. The performance of linear channel equalizers is poor especially when the nonlinear distortion is severe [2]. In these cases, nonlinear equalizers are preferable [1]-[6]. Since artificial neural network (ANN) can perform complex mapping between its input and output space, different ANNs have been successfully used in nonlinear inverse modeling problem [1]-[6]. Functional link artificial neural network (FLANN) possess a low complexity structure and has been employed as an adaptive inverse model [7]-[8]. Recently, genetic algorithm (GA) has been used for nonlinear and blind channel equalization [9]-[10]. The operations in GA such as the crossover and the mutation, help to avoid local minima problem and thus provide improved solution. However there are some situations when the weights in GA are trapped to local minima. The bacterial foraging optimization (BFO) [11] like GA is a derivative free optimization technique and acts as an useful alternative to GA. The number of parameters that are used for searching the total solution space is higher in BFO compared to those in GA but on the other hand requires less number of computations. The possibility of BFO being trapped to local minimum is less. In recent years the BFO has been proposed and has been applied to many areas such as harmonic estimation of power system signals [12] and adaptive inverse modeling [13]. In case of derivative free algorithms conventionally the mean square error (MSE) is used as the fitness or cost function. Use of MSE as cost function leads to improper training of the parameters of adaptive model when outliers are present in the desired signal. The traditional regressors employ least square fit which minimizes the Euclidean norm of the error, while the robust estimator is based on a fit which minimizes another rank based norm called Wilcoxon norm [14]. It is reported that linear regressors developed using Wilcoxon norm are robust against outliers. Using this new norm robust machines have recently been proposed for approximation of nonlinear functions [15]. Inverse modeling finds many applications such as channel equalization in telecommunication [8], adaptive linearization in intelligent sensors [16] and digital control [17]. In all these applications the learning tools used are derivative based such as least mean square (LMS) or back propagation (BP) algorithms. When outliers are present in the training signals the inverse modeling performance of these algorithms degrades substantially. To the best of our belief no literature has dealt on the problem of development of robust inverse model in presence of outliers. Therefore our motivation in this paper is to address this problem and to provide effective solution using BFO based training of the models using a robust norms of error [15, 18, 19] as cost function. The choice of BFO as a training tool is due to its advantages over other evolutionary computing tool stated earlier. The paper is organized into following sections : Section II discusses inverse modeling problem. Robust adaptive inverse model is given in Section III. In Section IV the BFO based update algorithm is developed for the inverse model. Development of robust inverse model using BFO based training and robust norm minimization is dealt in Section V. For performance evaluation, the simulation study is carried out and is dealt in Section VI. Finally conclusion of the paper is outlined in Section VII.

© 2011 ACEEE
DOI: 01.IJCSI.02.02.44

II. ADAPTIVE INVERSE MODEL

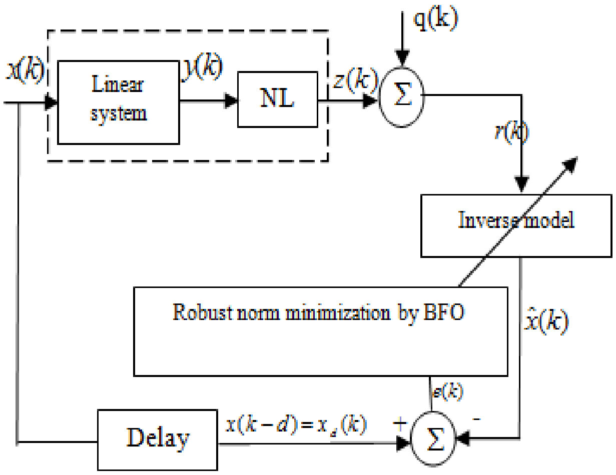


Fig.1. Development of a robust inverse model using BFO based training cost function

The input symbols are represented as $x(k)$ at time instance, k. They are then passed to a channel (plant) which may be linear or nonlinear. An FIR filter is used to model a linear channel whose output at time instant k may be written as

$$y(k) = \sum_{i=0}^{N-1} w(i)x(k-i) \tag{1}$$

where $w(i)$ are the weight values and N is the length of the FIR plant. The "NL" block represents the nonlinear distortion introduced in the channel and its output may be expressed as a nonlinear function of input and channel coefficients.

$$z(k) = \psi\,(x(k), x(k-1),........, x(k-N+1) \\ w(0), w(1),........ ...., w(N-1)), \tag{2}$$

where $\psi(.)$ is some nonlinear function generated by the "NL" block. The channel output $z(k)$ is corrupted with additive white Gaussian noise $q(k)$ of variance σ^2. This corrupted received output is given by $r(k)$. The received signal $r(k)$ is then passed into the adaptive inverse model to produce $\hat{x}(k)$ which represents the input symbols $x(k)$. From initial parameters, the weights are updated until the conventional cost function

$$CF_2 = \frac{1}{N}\sum_{k=1}^{N} e^2(k) \tag{3}$$

is minimized. The term N stands for input samples used for training and the error term $e(k) = x_d(k) - \hat{x}(k)$. The received data is given by $r(k) = z(k) + q(k)$. Conventionally the minimization of this cost function is performed iteratively by derivative based LMS algorithm.

III. ROBUST ADAPTIVE INVERSE MODEL

Three robust cost functions (RCF) defined in literature [15, 18, 19] are chosen in the development of robust adaptive inverse models. The RCFs are defined as

(a) Robust Cost Function-1 (Wilcoxon Norm) [14,15]

The Wilcoxon cost function is a pseudo-norm and is defined as

$$CF_1 = \sum_{i=1}^{l} a(R(v_i))v_i = \sum_{i=1}^{l} a(i)v_i \tag{4}$$

where $R(v_i)$ denotes the rank of v_i among $v_1, v_2, ..., v_l$, $v_{(1)} \le v_{(2)} \le ......v_{(l)}$ are the ordered values of $v_1, v_2, ...,v_l$, $a(i) = \phi[i/(l+1)]$. In statistics, different types of score functions have been dealt but the commonly used one is given by $\phi(u) = \sqrt{12}(u-0.5)$.

(b) Robust Cost Function-2 [18]

It is defined as

$$CF_3 = \sigma(1-\exp(-e^2/2\sigma)) \tag{5}$$

where σ is a parameter to be adjusted during training and e^2 is the mean square error

(c) Robust Cost Function-3 (Mean Log Squared error)[19]

The next cost function is defined as

$$CF_4 = \log(1+\frac{e^2}{2}) \tag{6}$$

where e^2 is mean square error.

The weight-update mechanism of inverse model of Fig. 1 is carried out by minimizing the cost functions of the errors defined in (4), (5) and (6) using BFO algorithm.

IV. DEVELOPMENT OF ROBUST NONLINEAR INVERSE MODELS USING BFO

The updating of the weights of the BFO based inverse model is carried out using the training rule as outlined in the following steps:

Step -1 **Initialization of parameters**

(i) S_b = No. of bacteria to be used for searching the total region

(ii) N = Number of input samples

(iii) p = Number of parameters to be optimized

(iv) N_s = Swimming length after which tumbling of bacteria is undertaken in a chemotactic loop.

(v) N_c =Number of iterations carried out in a chemotactic loop, $N_c >$.

© 2011 ACEEE
DOI: 01.IJCSI.02.02.44

(vi) = Maximum number of reproduction loop
(vii) = Maximum number of elimination and dispersal loop.

(viii) P_{ed} = Probability with which the elimination and dispersal takes place.

(ix) The location of each bacterium P (1- p , 1-S_b , 1) is specified by random numbers on [0,1].

(x) The value of run length unit, $C(i)$ is assumed to be constant for all bacteria.

Step-2 Generation of model input

(i) Random binary input [1,-1] is applied to the channel/plant.
(ii) The output of the channel is contaminated with white Guassian noise of known strength to generate the input signal.
(iii) The binary input is delayed by half of the order of the inverse model to obtain the signal, $x_d(k)$.

Step -3 Weight update algorithms

In this step the bacterial population, chemotaxis, reproduction, elimination and dispersal are modeled. Initially $j = k = l = 0$

(i) Elimination dispersal loop $l = l + 1$

(ii) Reproduction loop $k = k + 1$

(iii) Chemotaxis loop $j = j + 1$

(a) For $i = 1,2,...........S_b$, the cost function $J(i, j, k, l)$ for each i th bacterium is evaluated by the following way :

(1) N number of random binary samples are generated , passed through the model and the output is computed.
(2)The output is then compared with the corresponding training signal, $x_d(k)$ to calculate the error, $e(k)$.
(3)The robust cost functions of the error terms are computed as . $J(i, j, k, l)$
(4)End of For Loop.
(b)For $i = 1,2,..................,S_b$ the tumbling/swimming decision is taken.
Tumble : A random vecto $\Delta(i)$, with each element, $\Delta_m(i), m = 1,2,.........p,$ in the range of [-1, 1] is generated. The new position of i th bacterium in the direction of tumble is computed as

$$P^i(j+1,k,l) = P^i(j,k,l) + C(i) \times \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad (7)$$

The updated position is used to compute the new cost function (mean squared error) $J(i, j+1, k, l)$.
Swim – (i) Let c (counter for swim length) = 0
(ii) Whil $c < N_s$, hat is bacteria have not climbed down too long then update $c = c + 1$
If $J(j) < J(j-1)$ then the new position of th bacterium is computed by using (7). The updated position,
$P(j+1, k, l)$ is used to compute the new cost function,
$J(i, j+1, k, l)$

ELSE $c = N_s$. (End of the WHILE statement).

(c)Go to next bacterium $(i+1)$, if $i \neq S_b$ the process is repeated.

(d)If $\min(J)$ {minimum value of J among all the bacteria} is less than the tolerance limit then all loops are broken.

Step-4. If $j < N_c$,go to (iii) to continue chemotaxis loop since the lives of bacteria are not over.

Step-5 **Reproduction**

(a) For the given k and l, and for each $i = 1,2,..................,S_b$ the bacteria are sorted in ascending order of cost functions, J (higher cost function means lower health).
(b) The first half of bacteria population is allowed to die and the remaining half bacteria with least cost functions split into two and the copies are placed at the same location as their parent.

Step-6. If $k < N_{re}$ go to Step-2. In this case, the number of specified reproduction steps has not reached and the next generation in the chemotactic loop is started.

Step-7. **Elimination –Dispersal**
The bacterium with an elimination-dispersal probability above a preset value P_{ed} , is eliminated by dispersing to a random location and new replacements are randomly made in the search space. By this approach the total population is maintained constant.

V. Development of robust inverse model using BFO based training

Three robust cost functions defined in literature [15, 18, 19] are used to develop the robust inverse models. The BFO is then used to iteratively minimize these cost functions of the error obtained from the model. The resulting inverse model is expected to be robust against outliers. The weight-update of inverse model of Fig. 1 is carried out by minimizing these cost functions of the errors defined in (4), (5) and (6) using BFO algorithm.
In this approach, the procedure outlined in Steps-1 to 7 of Section IV remains the same. The only exception is detailed as follows :
Let the error vector of p th bacterium at k th generation due to application of N input samples to the model be represented as $[e_{1,p}(k), e_{2,p}(k),...........,e_{N,p}(k)]^T$.
The errors are then arranged in an increasing manner from which the rank $R\{e_{n,p}(k)\}$ of each n th error term is obtained. The score associated with each rank of the error term is evaluated as

$$a(i) = \sqrt{12}\left(\frac{i}{N+1} - 0.5\right) \qquad (8)$$

where $(1 \le i \le N)$ denotes the rank associated with each error term. At k th generation of each p th particle the Wilcoxon norm is then calculated as

© 2011 ACEEE
DOI: 01.IJCSI.02.02. 44

$$C_p(k) = \sum_{i=1}^{N} a(i)\, e_{i,p}(k) \qquad (9)$$

Similarly other two robust CFs are also computed using (5) and (6). The learning process continues until the CF decreases to the possible minimum values. At this stage the training process is terminated and the resulting weight vector represents the final weights of the inverse model

VI. SIMULATION STUDY

In this section, the simulation study of the proposed inverse model in presence of 10% to 50% of outliers in the training signal is carried out. Fig. 1 is simulated for various nonlinear channels using the algorithm given in Sections V and VI. The transfer functions of three standard linear systems used in the simulation study are :

$$\begin{aligned} H_1(z) &: 0.209 + 0.995z^{-1} + 0.209z^{-2} \\ H_2(z) &: 0.260 + 0.930z^{-1} + 0.260z^{-2} \\ H_3(z) &: 0.304 + 0.903z^{-1} + 0.304z^{-2} \end{aligned} \qquad (10)$$

The eigen-value-ratio (EVR) of the plants given in (10) are 6.08, 11.12 and 21.71 respectively [8] which indicate the complexity of the plant or channel. A higher EVR indicates a bad channel in the sense that the convergence of the model becomes poor. To study the effect of nonlinearity on the inverse model performance, two different nonlinearities are introduced

$$\begin{aligned} NL1 &: z(k) = \tanh(y(k)) \\ NL2 &: z(k) = y(k) + 0.2y^2(k) - 0.1y^3(k) \end{aligned} \qquad (11)$$

where $y(k)$ is the output of each of linear systems defined in (10). The additive noise in the channel is white Gaussian with -30dB strength. In this study an 8-tap adaptive FIR filter is used as an inverse model. The desired signal is generated by delaying the input binary sequence by half of the order of the inverse model. Outliers are added by simply replacing the bit value from 1 to -1 or -1 to 1 at randomly selected locations (10% to 50%) of the desired signal. In this simulation study, we have used the following parameters of BFO : S_b = 8, N_{is} = 100, p =8, N_s =3, N_c =5, N_{re} =40-60, N_{ed} =10, P_{ed} = 0.25, $C(i)$ = 0.0075. This selection of parameters is based on achieving best inverse modeling performance through simulation.

The bit-error-ratio (BER) plot of BFO trained inverse model pertaining to different nonlinear plants with different cost functions in presence of 0%-50% of outliers are obtained through simulation. The BER plots with 50% outliers only are shown in the Figs.2(a)-(f). In these figures the BER plots of four cost functions have been compared. To study the effects of EVR of the plant on BER performance the SNR is set at 15dB in presence of 0% and 50% outliers in the training signal and the results are shown in Fig. 3 for 50% outliers. Few notable observations obtained from these plots are :

(a)Keeping the CF, SNR and percentage of outliers in the training signal same, the BER increases with increase in the EVR of the channel. Similarly under identical simulation conditions the squared error cost function based inverse model performs the worst where as the Wilcoxon norm based model provides the best performance (least BER).

(b)As the outliers in the training signal increases the Wilcoxon norm based inverse model continues to yield lowest BER compared to those provided by other norms.

(c)With no outlier present in the training signal, the BER plot of all four CFs are almost same

(d)In presence of high outliers the conventional CF2 based model performs the worst followed by CF4 based model. In all cases the Wilcoxon norm (CF1) based inverse model performs the best and hence is more robust against low to high outliers in the training signal.

(e)The accuracy of inverse model based on CF3 and CF4 norms developed using outliers is almost identical.

(f)In addition, the plots of Fig. 3 indicate that at 50% outliers in the training signal the BER increases with increase in the EVR of the nonlinear plants.

(g) Further, the BER of the inverse models of all plants and SNR conditions is highest in square error norm (CF2) based training compared to three other robust norms used. However the Wilcoxon norm (CF1) based inverse model yields minimum BER among all cases studied.

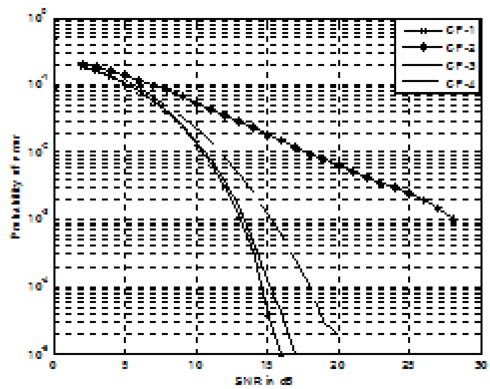


Fig. 2(a) Comparison of BER of four different CFs based nonlinear equalizers with [.209, .995, .209] as channel coefficients and NL1 with 50% outliers

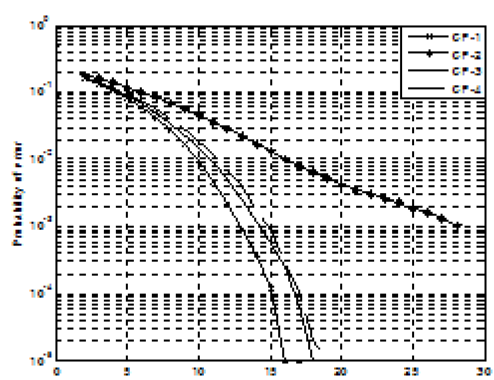


Fig.2(b) Comparison of BER of four different CFs based nonlinear equalizers with [.209, .995, .209] as channel coefficients and NL2 with 50% outliers

© 2011 ACEEE
DOI: 01.IJCSI.02.02. 44

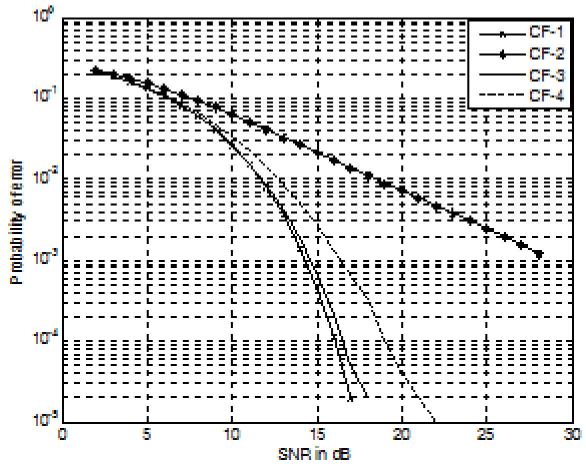


Fig.2(c) Comparison of BER of four different CFs based nonlinear equalizers with [.260, .930, .260] as channel coefficients and NL1 with 50% outliers

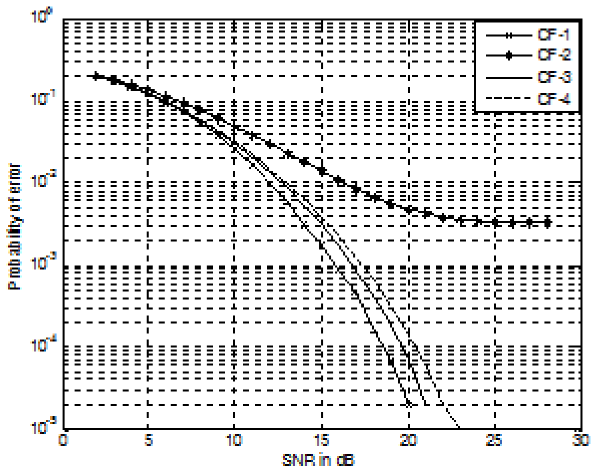


Fig. 2(d) Comparison of BER of four different CFs based nonlinear equalizers with [.260, .930, .260] as channel coefficients and NL2 with 50% outliers

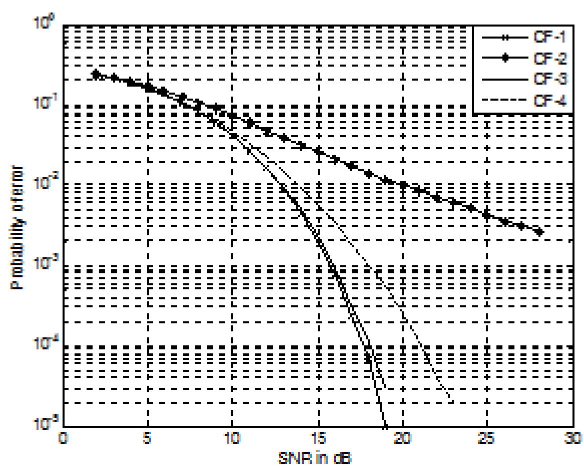


Fig. 2(e) Comparison of BER of four different CFs based nonlinear equalizers with [.304, .903, .304] as channel coefficients and NL1 with 50% outliers

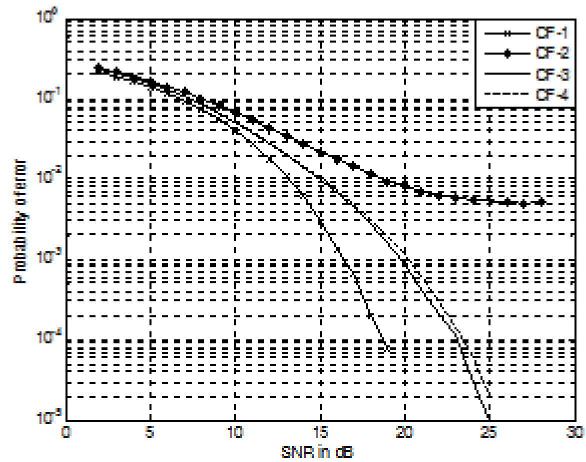
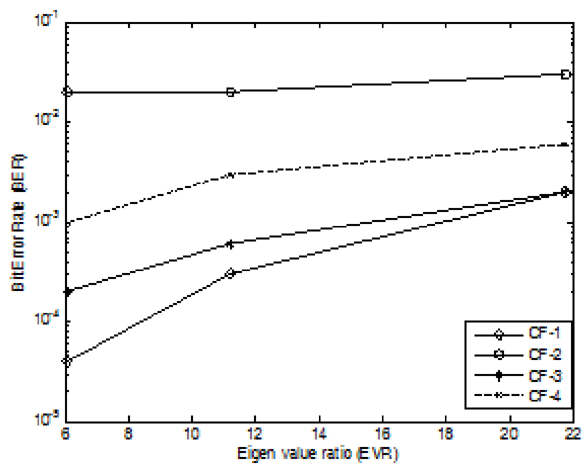
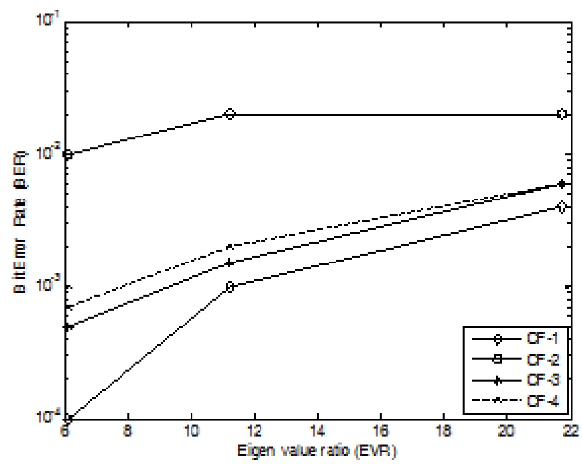


Fig.2(f) Comparison of BER of four different CFs based nonlinear equalizers with [.304, .903, .304] as channel coefficients and NL2 with 50% outliers



(a) NL1



(b) NL2

Fig. 3 Effect of EVR on the BER performance of the four CF-based equalizers in presence of 50% outliers

© 2011 ACEEE
DOI: 01.IJCSI.02.02. 44

ACEEE

VII. Conclusion

This paper examines and evaluates the learning capability of different robust norms of error when the training signal (of the inverse model) is contaminated with strong outliers. To facilitate such evaluation different nonlinear plants with varying EVRs are used. The population based BFO learning tool is developed to minimize four different norms. The robustness of these norms is assessed through extensive simulation study. It is in general observed that the conventional squared error norm (CF2) is least robust to develop inverse models of nonlinear systems under varying noise conditions whereas the Wilcoxon norm (CF1) is the most robust one. In terms of quality of performance, the norms are grouped in the order CF1, CF3, CF4 and CF2.

Acknowledgment

The work was supported by the Department of Science and Technology, Govt. of India under grant no. SR/S3/EECE/065/2008.

References

[1] S. Siu, G. J. Gibson and C. F. N. Cowan, “Decision feedback equalization using neural network structures and performance comparison with standard architecture, Proc. of Inst. Elect. Eng., vol. 137, pp. 221-225, 1990.

[2] S. Chen, G. J. Gibson, C. F. N. Cowan and P. M. Grant, “Adaptive equalization of finite nonlinear channels using multilayer perceptrons, EURASIP Journal of Signal Processing, vol. 20, pp. 107-119, 1990.

[3] M. Meyer and G. Pfeiffer, “Multilayer perceptron based decision feedback equalizers for channels with intersymbol interference”, IEE Proceeding, part-I, vol. 140, issue 6, pp. 420-424, 1993..

[4] S. Chen, B. Mulgrew and P. M. Grant, “A clustering technique for digital communication channel equalization using radial basis function networks”, IEEE Trans. on Neural Network, vol. 4,issue 4, pp. 570-579, 1993.

[5] P. Kumar, P. Saratchandran and N. Sundararajan, “Minimal radial basis function neural networks for nonlinear channel equalization”, IEE Proc. Vis. Image Signal Processing, vol. 147, pp. 428-435, 2000.

[6] J. Lee and R. Sankar, “Theoretical derivation of minimum mean square error of RBF based equalizer”, Signal Processing, vol. 87, pp. 1613-1625, 2007.

[7] J. C. Patra, Wei Beng Poh, N. S. Chaudhari and Amitabha Das, “Nonlinear channel equalization with QAM signal using Chebyshev artificial neural network”, Proc. of International joint conference on neural networks, Montreal, Canada, pp. 3214-3219, August 2005.

[8] J. C. Patra, R. N. Pal, R. Baliarsingh and G. Panda, “Nonlinear channel equalization for QAM signal constellation using Artificial Neural Network”, IEEE Trans. on systems, man and cybernetics-Part B:cybetnetics, vol. 29, no. 2, April 1999.

[9] Saman S. Abeysekera, “Adaptive blind channel equalization using orthogonalization and plane rotations via the Genetic Algorithm”, International Conference on Communications, circuits and systems, vol. 2, pp.895-899, 27-30 May 2005.

[11] K. M. Passino, “Biomimicry of Bacterial Foraging for distributed optimization and control”, IEEE control system magazine, vol 22, issue 3, pp. 52-67, June 2002.

[10] G. Panda, Babita. Majhi, D. Mohanty, A. Choubey and S. Mishra, “Development of Novel Digital Channel Equalizers using Genetic Algorithms”, Proc. of National Conference on Communication (NCC-2006), IIT Delhi, pp.117-121, 27-29,January, 2006.

[12] S. Mishra, “A Hybrid least square Fuzzy bacterial foraging strategy for harmonic estimation”, IEEE Trans. on Evolutionary Computation, vol 9, no. 1, pp. 61-73, Feb. 2005.

[13] Babita Majhi, G. Panda and A. Choubey, “On The Development of a new Adaptive Channel Equalizer using Bacterial Foraging Optimization Technique”, Proc. of IEEE Annual India Conference (INDICON-2006), New Delhi, India, 15th-17th September, 2006, pp. 1-6.

[14] Joseph W. McKean, “Robust analysis of Linear models”, Statistical Science, vol. 19, no. 4, pp. 562-570, 2004.

[15] Jer-Guang Hsieh, Yih-Lon Lin and Jyh-Horng Jeng, “Preliminary study on Wilcoxon learning machines”, IEEE Trans. on neural networks, vol.19, no. 2, pp. 201-211, Feb. 2008.

[16] J. C. Patra, A. C. Kot and G. Panda, “An intelligent pressure sensor using neural networks”, IEEE Trans. on Instrumentation and Measurement, vol. 49, issue 4, pp. 829-834, Aug. 2000.

[17] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks”, IEEE Trans. on Neural Networks, vol. 1, pp. 4-26, January 1990.

[18] Wei-Yen Wang, Tsu-Tian Lee, Ching-Lang Liu and Chi-Hsu Wang, “Function approximation using fuzzy neural networks with robust learning algorithm”, IEEE Trans. on Systems, Man and Cybernetics-Part B : Cybernetics, vol. 27, no. 4, pp. 740-747, Aug. 1997.

[19] Hung-Hsu Tsai and Pao-Ta Yu, “On the optimal design of fuzzy neural networks with robust learning for function approximation”, IEEETrans.on Systems, Man and Cybernetics-Part B : Cybernetics, vol. 30, no. 1, pp. 217-223, Feb. 2000.

© 2011 ACEEE
DOI: 01.IJCSI.02.02. 44